

# Um Modelo de Estimação para Provisionamento de Rede de Máquinas Virtuais em Nuvens IaaS

Jonatas A. Marques<sup>1</sup>, Rafael R. Obelheiro<sup>1</sup>

<sup>1</sup> Programa de Pós-Graduação em Computação Aplicada  
Universidade do Estado de Santa Catarina, Joinville, Brasil

jonatas.marques@ufrrgs.br, rafael.obelheiro@udesc.br

**Abstract.** *In Infrastructure-as-a-Service (IaaS) clouds, customers lease virtual resources (e.g., CPU, memory, network) offered by cloud providers, paying for the reserved capacity of resources, regardless of their effective use. In this context, it is in the interest of customers to reserve resources with sufficient capacity so that their applications achieve good performance whilst, at the same time, seeking to minimize their expenses with idle capacities. In this paper, we present a novel model for network bandwidth estimation to guide cloud users in the provisioning of their virtual machines. The model is application-agnostic and dynamically adapts to changes in the virtual machine's workload profile through automatic monitoring and classification of its network flows. Our experimental evaluation shows that the estimates provided by the model are useful for guiding the provisioning of virtual machines, providing accurate bandwidth estimates even when the network is currently under-provisioned.*

**Resumo.** *Em nuvens computacionais do tipo Infrastructure-as-a-Service (IaaS), clientes alugam recursos virtuais (processador, memória, rede) fornecidos por provedores de nuvem, pagando pela capacidade reservada, independente da efetiva utilização dos recursos. Nesse contexto, é de interesse dos clientes reservar recursos com capacidade suficiente para que suas aplicações atinjam um bom desempenho, buscando ao mesmo tempo minimizar gastos com capacidade ociosa. Neste trabalho, é proposto um novo modelo para estimação de largura de banda de rede adequada para provisionamento de máquinas virtuais em nuvens IaaS, considerando a perspectiva do cliente. O modelo é agnóstico de aplicação e adapta-se dinamicamente à evolução do perfil de carga da máquina virtual, sendo baseado na monitoração e classificação automática dos seus fluxos de rede. A avaliação experimental mostra que as estimativas fornecidas pelo modelo são úteis para orientar o provisionamento de máquinas virtuais, sendo capaz de gerar estimativas acuradas de largura de banda necessária mesmo em cenários onde a rede está subprovisionada.*

## 1. Introdução

A computação em nuvem é um modelo que permite o acesso a recursos computacionais configuráveis (como redes, servidores, armazenamento, aplicações e serviços) por seus clientes [Mell and Grance 2011], que pagam conforme a capacidade reservada dos recursos, independente da efetiva utilização destes [Suleiman et al. 2012]. Nesse contexto, deve-se ressaltar que (i) o cliente deve especificar a capacidade desejada para cada recurso

individualmente e (ii) em nuvens, capacidade em excesso induz um custo desnecessário com o aluguel de recursos ociosos (diferente de infraestruturas físicas onde o custo de aquisição de recursos é fixo), portanto o cliente deseja reservar a capacidade mínima que ainda proporcione um bom desempenho às suas aplicações.

De modo geral, em nuvens IaaS (*Infrastructure-as-a-Service*) a rede é compartilhada entre os clientes, sem garantia da largura de banda (LB) disponível para as máquinas virtuais. Essa ausência de garantia prejudica não apenas os clientes, pois o desempenho de muitas aplicações é sensível ao desempenho da rede e este se torna imprevisível [Ballani et al. 2011], mas também os provedores, que podem arcar com custos decorrentes de violações de acordos de nível de serviço (*service level agreements*, SLAs) e deixar de atrair clientes que demandem previsibilidade da rede [Mogul and Popa 2012]. Abstrações mais refinadas, tais como infraestruturas virtuais [Anhalt et al. 2010], permitem reservar LB para enlaces virtuais, mas exigem que o cliente especifique a capacidade desejada, o que não é uma tarefa trivial. Portanto, um mecanismo que seja capaz de estimar a LB adequada para enlaces virtuais seria útil tanto para os clientes, que poderiam procurar serviços com garantia da capacidade de rede, quanto provedores, pois facilitaria a oferta de serviço garantido com bom custo-benefício ao permitir uma utilização otimizada das infraestruturas físicas com mínimo risco de problemas de desempenho.

O dimensionamento de enlaces virtuais para ambientes de nuvem ainda é pouco explorado na literatura. Alguns trabalhos focam em mecanismos para oferecer garantias de LB em nuvens, com base em demandas especificadas pelos clientes [Popa et al. 2012, Anastasi et al. 2016, Marcon et al. 2016]. Em [Pfischer et al. 2013] é proposto um modelo de diagnóstico de provisionamento de rede que identifica se a LB está subprovisionada, superprovisionada ou adequada, mas que, nos casos de sub- e superprovisionamento, não estima a LB adequada. Cicada [LaCurts et al. 2014] propõe um mecanismo para predição de LB entre pares de MVs considerando a demanda média ou de pico em um dado intervalo, mas não trata o tráfego entre a MV e *hosts* Internet e nem é capaz de medir acuradamente a LB necessária quando a rede está subprovisionada.

Visando a contornar essas limitações, este trabalho traz como contribuição uma nova abordagem para estimação da LB adequada para enlaces virtuais. As interfaces de rede de uma MV são monitoradas para observar o volume de dados transmitido por todas as aplicações, sem a necessidade de conhecê-las. Algoritmos de aprendizado de máquina são usados para identificar grupos de fluxos com demanda de rede semelhante, e o tráfego de rede é classificado segundo esses grupos para estimar a LB adequada para essa MV. Resultados experimentais mostram que essa abordagem fornece estimativas suficientemente acuradas mesmo quando a rede está subprovisionada.

O restante deste artigo está organizado como segue. A Seção 2 discute os trabalhos relacionados. A Seção 3 descreve o modelo proposto, e a Seção 4 apresenta sua avaliação experimental. Finalmente, a Seção 5 conclui o artigo.

## 2. Trabalhos Relacionados

Cicada [LaCurts et al. 2014] é um mecanismo para predição da LB de enlaces virtuais sob o modelo *pipe* [Mogul and Popa 2012], em que são especificadas as capacidades dos enlaces entre pares de MVs. A monitoração do tráfego se baseia na análise dos fluxos entre MVs em um *datacenter*, sendo explicitamente desconsiderado o tráfego entre MVs

e a rede externa (clientes na Internet). A matriz de tráfego é observada a intervalos regulares, e a LB prevista é dada por uma combinação linear de observações anteriores, com pesos que vão sendo ajustados automaticamente em função dos erros de previsão. O mecanismo pode prever tanto a LB média quanto a LB de pico durante um dado período. O trabalho adota a premissa de que a rede está suficientemente provisionada durante o período de monitoração, não tratando o caso em que ela está subprovisionada. Cicada também incorpora um algoritmo para mapeamento dos recursos virtuais na infraestrutura física. O presente trabalho tem um escopo mais estreito, concentrando-se no problema de estimação da LB adequada para uma dada MV, mas considera o tráfego externo e é capaz de fornecer estimativas acuradas mesmo quando a rede está subprovisionada. Adaptar nossa abordagem para o modelo *pipe* seria uma tarefa trivial, bastando separar os fluxos de acordo com os seus pares origem-destino.

Em [Pfitscher et al. 2013] é proposto um modelo de diagnóstico de provisionamento de rede capaz de identificar se a largura de banda (LB) está subprovisionada, superprovisionada ou adequada. Esse modelo é baseado na monitoração do consumo de banda e da fila das interfaces de rede de uma máquina virtual (MV). Apesar do modelo ser capaz de identificar o estado de provisionamento de LB, ele não estima a LB nos casos de sub- e superprovisionamento. O presente trabalho vale-se da monitoração de fluxos para estimar a LB necessária em uma MV. Essa estimativa pode ser comparada com a LB alocada para diagnosticar o provisionamento de rede.

Alguns trabalhos, como [Gong et al. 2010, Engelbrecht and van Greunen 2015, Nguyen et al. 2013, Moreira Jr and Obelheiro 2016], propuseram estratégias de previsão do nível de utilização de recursos. Esses trabalhos são baseados em algoritmos de regressão (e.g., auto-regressão, auto-correlação, ARMA, ARIMA, suavização exponencial) que consideram históricos de medições de utilização de recursos com o objetivo de prever valores futuros. Em contraste com o presente trabalho, esses trabalhos focam no provisionamento adequado de processador e memória. Além disso, nenhuma dessas estratégias consideram o impacto do subprovisionamento na medição da utilização dos recursos. O modelo proposto no presente trabalho poderia ser usado em conjunto com uma das estratégias propostas nesses trabalhos para melhorar a qualidade do histórico de consumo de LB de rede para previsão de necessidades futuras.

[Popa et al. 2012, Anastasi et al. 2016, Marcon et al. 2016] propuseram mecanismos e estratégias para oferecer garantias de LB em nuvens. Esses trabalhos assumem que os clientes de nuvens especificam a LB desejada no momento da reserva de recursos. Nossa proposta é ortogonal a esses trabalhos e os complementa auxiliando os clientes de nuvem na tarefa de determinar qual a LB adequada para que suas aplicações atinjam bom desempenho sem que a reserva incorra em custos supérfluos.

### 3. Modelo Proposto

O fluxo de execução do modelo proposto é mostrado na Figura 1, e suas etapas são detalhadas na sequência. O modelo produz uma estimativa da largura de banda necessária para a carga aplicada sobre uma máquina virtual em um dado período. Estimar a LB necessária é um problema particularmente difícil quando a rede está subprovisionada, pois não é possível observar diretamente uma demanda superior à capacidade alocada. Nosso modelo traz como contribuição uma nova abordagem que permite estimar indiretamente

o valor real da demanda quando a rede está subprovisionada.

As estimativas geradas pelo modelo não são, por si sós, previsões da LB necessária, mas em conjunto podem ser utilizadas por um algoritmo de previsão para esse fim. No presente trabalho, intui-se que a qualidade das estimativas de LB necessária passadas tem impacto significativo sobre a acurácia das previsões futuras. Usar as estimativas produzidas em uma sequência de períodos para obter previsões da LB necessária futura enquadra-se no escopo de outro trabalho em andamento.

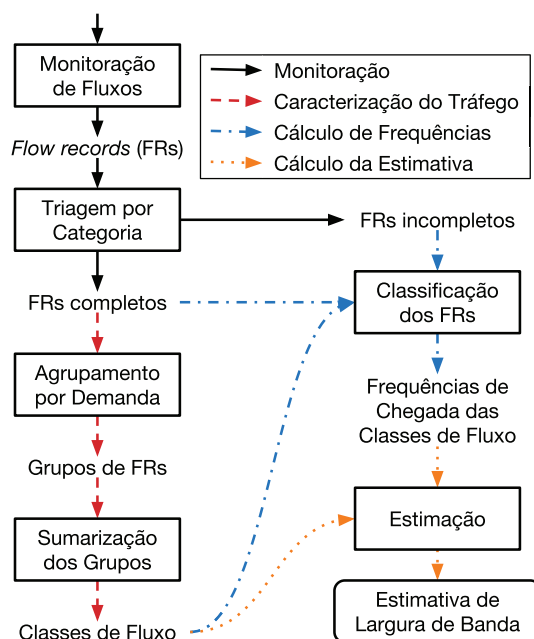


Figura 1. Fluxo de execução do modelo proposto.

### 3.1. Monitoração de Fluxos

O fluxo de execução do modelo proposto inicia pela monitoração de fluxos nas interfaces de rede da máquina virtual (MV) da qual deseja-se estimar a largura de banda (LB) necessária. Neste trabalho, essa monitoração é realizada usando o Argus<sup>1</sup>. Após um período de monitoração, o Argus fornece um relatório do tráfego que consiste em um conjunto de *flow records* (FRs), sendo registrado um FR para cada fluxo (bidirecional) observado. Um FR consiste em um registro que traz diversas informações sobre um fluxo. Alguns exemplos dessas são: os instantes inicial e final em que o fluxo foi observado; os endereços de origem e destino das aplicações comunicantes (i.e., endereço IP e porta); o protocolo de rede (e.g., UDP e TCP); as *flags* que foram observadas em pelo menos um dos pacotes do fluxo (e.g., SYN, ACK, FIN); e a quantidade de pacotes e bytes transmitidos em cada sentido do fluxo.

Realizada a monitoração de fluxos, pode-se então realizar a caracterização do tráfego das aplicações. Caracterizar o tráfego das aplicações consiste em identificar quais são as diferentes demandas de rede típicas dos fluxos das aplicações. A demanda de rede de um fluxo é expressa pelo par:

$$\text{Demanda} = \{\text{quantidade de bytes enviados, quantidade de bytes recebidos}\} \quad (1)$$

<sup>1</sup><http://www.qosient.com/argus/>

Par esse formado pelos valores da quantidade de bytes transmitido em seus dois sentidos. A hipótese do modelo proposto é que conhecendo as demandas típicas dos fluxos e a frequência com que fluxos de cada demanda ocorrem na MV, pode-se estimar a LB necessária da rede para que minimizar a ocorrência de filas nas interfaces de rede da MV. Visto que a formação de filas causa atraso na transmissão (e eventualmente a perda) de pacotes, prejudicando o desempenho da aplicação.

### 3.2. Triagem por Categoria

Os *flow records* (FRs) gerados pela monitoração podem ser categorizados como completos ou incompletos. Um FR é dito completo quando traz informações referentes a um fluxo que iniciou e terminou dentro do intervalo de monitoração. Quando um fluxo ocorre (em parte) fora do intervalo de monitoração (i.e., iniciou antes e/ou terminou após o período de monitoração), o FR que traz informações sobre este fluxo é dito incompleto, pois durante o período de monitoração observou-se apenas parte de sua demanda de rede. Para a finalidade de caracterização do tráfego apenas os FRs completos são de interesse, visto que somente esses possuem informações totais sobre a demanda de rede dos fluxos.

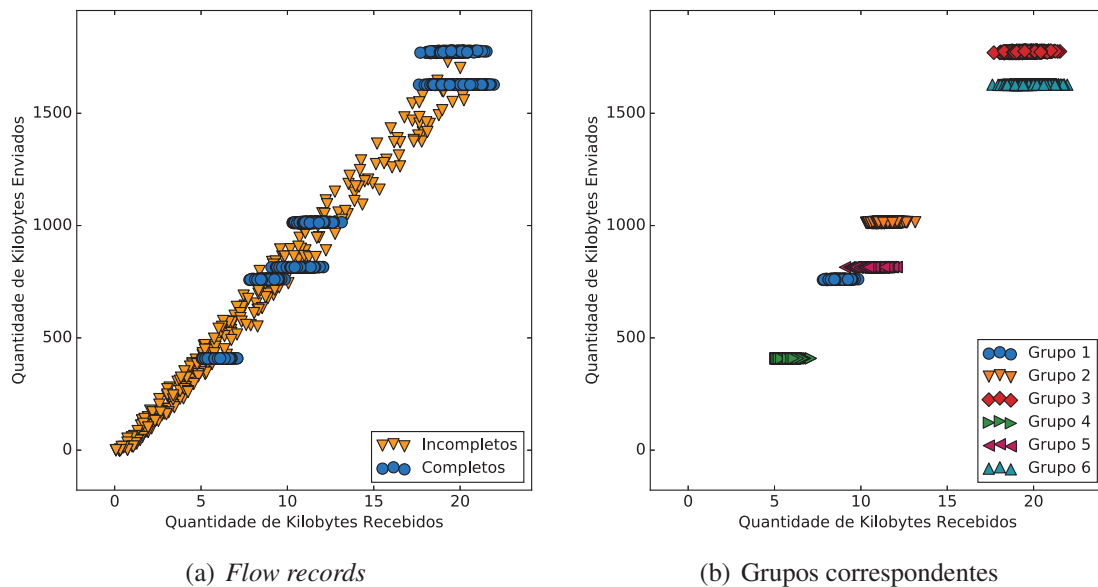
A etapa de triagem categoriza os FRs e os separa entre completos e incompletos. Os FRs de fluxos TCP são categorizados pelas *flags* observadas nos pacotes do fluxo. Caso as *flags* SYN (que indica o início de uma sessão TCP) e FIN (que indica o fim de uma sessão TCP) tenham sido usadas em alguns dos pacotes do fluxo, esse é categorizado como completo. Noutro caso, em que apenas uma ou nenhuma das duas *flags* seja observada, o fluxo TCP é categorizado como incompleto. Se a *flag* SYN não foi observada, conclui-se que o fluxo teve início anterior ao período de monitoração. Se a *flag* FIN não foi observada, conclui-se que o fluxo continuou além do período de monitoração ou foi abortado. Por sua vez, os FRs de fluxos UDP são todos categorizados como completos, visto que esse protocolo não apresenta sessões e todo datagrama transmitido representa uma transação/mensagem completa.

A Figura 2(a) apresenta a categorização do processo de triagem sobre um conjunto de FRs. Observa-se que os FRs completos formam grupos bem caracterizados, enquanto os incompletos estão espalhados ao longo da linha formada entre a origem do gráfico e os grupos de FRs completos. Em virtude desta observação, a fase de caracterização do tráfego considera apenas os FRs completos.

### 3.3. Agrupamento por Demanda

As aplicações voltadas ao atendimento de requisições (e.g., servidores de páginas web, bancos de dados) normalmente possuem um conjunto finito de requisições que são realizadas regularmente. A fase de caracterização do tráfego tem por objetivo identificar as demandas de rede dessas requisições observando unicamente os fluxos transitando nas interfaces de rede, sem a necessidade de entender o funcionamento das aplicações ou o que cada requisição representa para a aplicação. Essa metodologia permite ao modelo ser agnóstico às aplicações executando na MV.

Por experimentação, observou-se que requisições idênticas para uma mesma aplicação podem apresentar variações de quantidade de bytes recebidos e enviados (Figura 2(a)). Isso ocorre devido a fatores como o encapsulamento de mensagens de aplicação em segmentos TCP, perdas, reordenação de pacotes e ACKs atrasados (*delayed*



**Figura 2. Fluxos observados e grupos identificados para uma carga de trabalho.**

ACKs). Essas variações, no entanto, não têm magnitude suficiente para descaracterizar a demanda das requisições.

Para administrar essas variações de demanda para requisições idênticas, o modelo proposto aplica a técnica de *clustering* para identificar grupos de fluxos (que representam requisições) com demandas de rede próximas. A etapa de agrupamento busca identificar os grupos de fluxos com demandas de rede diferentes para cada uma das aplicações executando na MV. Essa etapa é realizada em duas partes. Primeiramente, os *flow records* (FRs) são agrupados quando possuem exatamente os mesmos endereços IP e portas de destino e protocolos de rede. No modelo proposto, cada tripla única [endereço IP de destino, porta de destino, protocolo de rede] representa uma “aplicação” diferente. Por exemplo, uma aplicação servidora Apache<sup>2</sup> com um *virtual host* configurado para atender a porta 80 e outro para atender a porta 443 é considerado pelo modelo como duas aplicações diferentes, pois utiliza portas diferentes.

Após o agrupamento por aplicação, cada grupo de fluxos de aplicação é subdividido (por um algoritmo de *clustering*) em grupos de fluxos com demanda de rede semelhante. Dessa forma, para cada requisição de cada aplicação é identificado indiretamente um grupo de fluxos. O algoritmo de *clustering* escolhido para implementação do protótipo do modelo proposto foi o Mini-Batch K-Means, disponível através da biblioteca SciKit-Learn [Pedregosa et al. 2011]. Mini-Batch K-Means apresentou, quando comparado com outros cinco algoritmos, o melhor compromisso entre complexidade de ajuste, complexidade de tempo e representatividade de agrupamento. O número de grupos usado no algoritmo é selecionado automaticamente, de forma iterativa: o algoritmo é executado várias vezes com diferentes números de grupos, e o melhor agrupamento – aquele que minimiza a variância da distância intragrupo – é o escolhido. A Figura 2(b) mostra os grupos identificados automaticamente pela etapa de agrupamento dos FRs da Figura 2(a).

<sup>2</sup><https://httpd.apache.org/>



Como comentado anteriormente, essa etapa considera apenas os *flow records* categorizados como completos. A consideração dos incompletos acarretaria na formação de grupos que não representam adequadamente as demandas totais típicas da aplicação. Considerar somente os *flow records* completos (durante o agrupamento) é importante principalmente em casos onde a rede está subprovisionada, pois nesses, o percentual de *flow records* incompletos dentre todos os observados costuma ser maior do que em casos de provisionamento adequado e superprovisionamento. Logo, a triagem proposta nesse trabalho é essencial para obter grupos de fluxos com demandas representativas e boa acurácia na estimação de LB.

### 3.4. Sumarização dos Grupos

Esta última etapa da fase de caracterização do tráfego tem por objetivo criar protótipos para os grupos de *flow records* (FRs) identificados pela etapa de agrupamento. Para cada grupo de FRs é criado um protótipo que o resume. Um protótipo, ou doravante chamado classe de fluxo, é composto de sete informações, sendo as três primeiras a tripla de identificação de uma aplicação: endereço IP de destino, porta de destino e protocolo de rede. As outras quatro informações expressam a demanda de rede típica da classe de fluxo, são estas: a quantidade média de bytes enviados em cada fluxo, a margem de erro da quantidade de bytes enviados em cada fluxo (com nível de confiança de 99%), a quantidade média de bytes recebidos em cada fluxo e a margem de erro da quantidade de bytes recebidos em cada fluxo (com nível de confiança de 99%).

A Tabela 1 exemplifica o resultado da etapa de sumarização e da fase de caracterização do tráfego sobre os grupos de FRs da Figura 2(b). A partir dessa tabela (e das classes por ela apresentada) conclui-se que existem duas aplicações em execução na MV sendo monitorada, uma operando na porta 8000 e outra na porta 9000. A primeira aplicação (porta 8000) atende a requisições com demandas de envio típicas de aproximadamente 761 KB, 1 MB e 1,8 MB. A outra aplicação (porta 9000) atende a requisições com demandas de envio típicas de aproximadamente 410 KB, 818 KB e 1,63 MB.

**Tabela 1. Classes de Fluxo geradas pela etapa de sumarização dos grupos.**

Classe	Endereço IP de Destino	Porta de Destino	Protocolo de Rede	Média Bytes Enviados ± Margem de Erro	Média Bytes Recebidos ± Margem de Erro
1	1.1.1.1	8000	TCP	761.070 ± 352	9.727 ± 213
2	1.1.1.1	8000	TCP	1.013.472 ± 600	12.040 ± 238
3	1.1.1.1	8000	TCP	1.775.569 ± 960	22.170 ± 562
4	1.1.1.1	9000	TCP	409.906 ± 167	7.330 ± 150
5	1.1.1.1	9000	TCP	817.707 ± 461	13.326 ± 410
6	1.1.1.1	9000	TCP	1.633.532 ± 404	25.436 ± 379

### 3.5. Classificação dos *Flow Records*

As etapas descritas nas subseções anteriores ajustam parcialmente o modelo através das classes de fluxo que caracterizam o tráfego das aplicações em execução na MV. Para estimar a LB necessária é essencial também identificar com que frequência fluxos de cada uma das classes iniciam nas interfaces de rede. Normalmente, a frequência de chegada de fluxos de cada classe é mais dinâmica do que a demanda de rede dessas, visto que é comum essa primeira variar ao longo de um dia (e.g., popularidade das páginas) enquanto a variação dessa segunda costuma estar atrelada a modificação da aplicação (e.g., criação de nova página em um servidor de páginas web). Para identificar a frequência de chegada

de cada classe de fluxo o modelo proposto realizada uma etapa de classificação sobre todos (completos e incompletos) os *flow records* (FRs) observados pela monitoração às classes.

Como comentado na Seção 3.2, os FRs fornecidos no relatório da monitoração de fluxos podem ser categorizados como completos ou incompletos. Para a finalidade de caracterização do tráfego, apenas os FRs considerados completos são observados, visto que somente estes contêm informações totais de demanda de rede. Já para a finalidade de identificar as frequências de chegada das classes de fluxo, ambas as categorias dos FRs devem ser consideradas (ainda que de formas diferentes). A classificação dos FRs completos associa cada um deles à classe de fluxo (de mesma aplicação, considerando os campos de identificação de aplicação) com demanda de rede mais próxima segundo a Distância Euclidiana, calculada através da Equação 2:

$$d_{i,a} = \sqrt{(R_a - R_i)^2 + (E_a - E_i)^2} \quad (2)$$

Sendo  $d_{i,a}$  a distância entre um FR  $i$  e uma classe de fluxo  $a$ ;  $R_a$  e  $E_a$  a quantidade média de bytes recebidos e enviados, respectivamente, da classe  $a$ ;  $R_i$  e  $E_i$  a quantidade de bytes recebidos e enviados, respectivamente, pelo fluxo  $i$ .

A classificação dos FRs incompletos, por sua vez, não pode ser baseada na distância euclidiana entre as demandas de um fluxo e das classes, pois os FRs incompletos contêm uma demanda de rede parcial do fluxo (apenas a quantidade de bytes observada durante a monitoração). Por exemplo, é possível que um fluxo com demanda total igual a [1 MB, 10 KB] seja observado apenas pela metade, resultando em um FR incompleto com demanda parcial de aproximadamente [500 KB, 5 KB]. Se o tráfego for caracterizado por três classes com demandas médias iguais a (a) [100 KB, 1 KB], (b) [600 KB, 6 KB] e (c) [1 MB, 10 KB], não é possível determinar categoricamente que quando observado por completo o FR pertenceria a classe (b) ou (c). Entretanto, como a demanda parcial registrada no FR tem valor superior à demanda da classe (a), conclui-se que o fluxo completo não pertenceria a essa classe.

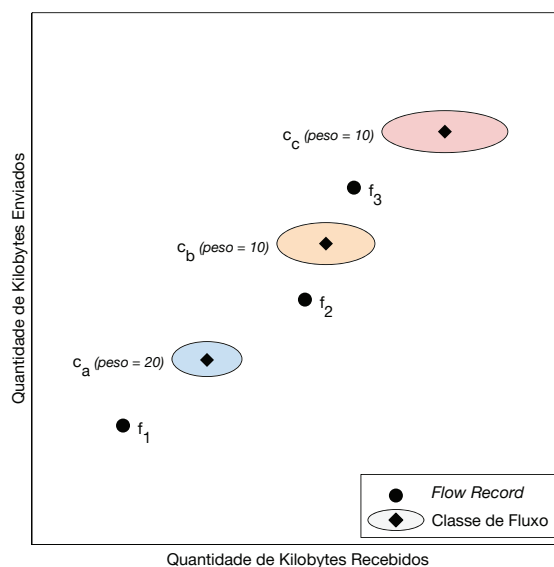


Figura 3. Exemplo de cenários de classificação de *flow records* incompletos.



Diante dessa problemática, o modelo proposto classifica FRs incompletos usando três cenários (otimista, pessimista e intermediário), e faz uma composição das três classificações para chegar à classificação final. No primeiro cenário, cada FR incompleto é classificado como pertencente à classe de mesma aplicação com demanda de rede imediatamente superior à demanda parcial do fluxo. Esse cenário otimista representa a LB mínima que seria necessária para transmitir completamente todos os fluxos observados parcialmente, pois cada fluxo é associado à classe de menor demanda dentre as classes com demanda superior à do fluxo. Nesse cenário os FRs incompletos  $f_1$ ,  $f_2$  e  $f_3$  da Figura 3 seriam atribuídos às classes  $c_a$ ,  $c_b$  e  $c_c$ , respectivamente.

O segundo cenário considerado classifica todos os FRs incompletos como pertencentes à classe de maior demanda da mesma aplicação. Esse cenário pessimista representa a LB máxima que seria necessária para transmitir completamente todos os fluxos observados parcialmente, pois todo fluxo é considerado como pertencente à classe de demanda de rede máxima da aplicação. Neste cenário todos os FRs incompletos da Figura 3 seriam atribuídos à classe  $c_c$ .

O terceiro (e último) cenário considerado associa parcelas dos *flow records* incompletos às classes. A parcela referente a cada classe é distribuída segundo uma ponderação feita sobre a frequência de chegada de fluxos completos de cada classe:

$$p_a = \frac{\lambda_a^c}{\sum_{i \in C_{>}} \lambda_i^c} \quad (3)$$

Sendo  $p_a$  o peso da classe  $a$ ;  $\lambda_a^c$  a frequência de chegada de fluxos completos da classe  $a$ ;  $C_{>}$  o conjunto de todas as classes de fluxo com demanda superior à do fluxo sob classificação. Neste cenário, à classe  $c_a$  da Figura 3 seria atribuído 1/2 do fluxo  $f_1$ ; à classe  $c_b$  seria atribuído 1/4 do fluxo  $f_1$  e 1/2 do fluxo  $f_2$ ; e à classe  $c_c$  seria atribuído 1/4 do fluxo  $f_1$ , 1/2 do fluxo  $f_2$  e todo o fluxo  $f_3$ . Esse cenário intermediário representa a LB que seria necessária para transmitir completamente todos os fluxos observados parcialmente caso a proporção entre as frequências de chegada observadas das classes fosse a mesma para os fluxos completos e para os incompletos.

**Tabela 2. Frequências de chegada das classes de fluxo do exemplo.**

Classe	<i>Flow records</i> completos (un.)	<i>Flow records</i> incompletos (un.)	Taxa de Chegada (fluxos/s)
1	436	28,85	3,1
2	719	17,94	4,91
3	697	51,88	4,99
4	1.318	70,13	9,25
5	419	25,15	2,96
6	1.206	132,72	8,92

As frequências de chegada das classes de fluxo considerando os FRs incompletos são calculadas pela média aritmética das frequências calculadas nos três cenários discutidos. Como comentado anteriormente na Seção 3.3, o percentual de FRs incompletos dentre todos os observados costuma ser significativo nos casos de subprovisionamento. Portanto, a classificação correta dos FRs incompletos (além dos completos) é fundamental para estimar a largura de banda necessária nos casos de subprovisionamento de rede. As frequências de chegada totais das classes de fluxo são calculadas somando as frequências

calculadas considerando os FRs completos e as calculadas considerando os incompletos. A Tabela 2 mostra as frequências de chegada de fluxos calculadas para as classes da Tabela 1 pela etapa de classificação dos fluxos.

### 3.6. Estimação

Após o ajuste completo do modelo proposto através dos processos (i) de caracterização do tráfego (ii) e de cálculo das frequências de chegada das classes de fluxo, estima-se a LB necessária para as aplicações em execução na MV segundo a Equação 4:

$$LB_{\text{necessária}} = \max\left(\sum_{i \in C} \lambda_i E_i; \sum_{i \in C} \lambda_i R_i\right) \quad (4)$$

Sendo  $C$  o conjunto de todas as classes de fluxo;  $\lambda_i$  a frequência de chegada de fluxos da classe  $i$ ;  $E_i$  a quantidade média de bytes enviados pela MV nos fluxos da classe  $i$ ; e  $R_i$  a quantidade média de bytes recebidos pela MV nos fluxos da classe  $i$ . A LB necessária para atender as classes da Tabela 2, por exemplo, é estimada como 295,3 Mbps (máximo entre 295,3 Mbps de envio e 3,6 Mbps de recepção).

Para garantir o bom desempenho das aplicações, durante o provisionamento de rede da MV recomenda-se a adição de uma folga à LB necessária para tolerar oscilações naturais da taxa instantânea de transmissão de rede. Neste trabalho recomenda-se (após observação experimental) provisionar a rede de forma que a LB necessária use em média entre 75% e 85% da capacidade. Por exemplo, para a estimativa de 295,3 Mbps, a LB a ser provisionada está entre 347,4 Mbps (85%) e 393,7 Mbps (75%). Essa folga pode ser ajustada pelo cliente de nuvem de modo a privilegiar o custo (reduzindo a folga) ou o desempenho (aumentando a folga).

## 4. Avaliação

### 4.1. Variação da Popularidade de Requisições

Este primeiro experimento tem por objetivo avaliar a qualidade da fase de caracterização de demanda de rede do tráfego e a acurácia da etapa de classificação de fluxos. Para este propósito, um servidor de páginas web foi configurado para atender requisições a 25 páginas de tamanhos e conteúdos distintos.

**Tabela 3. Tamanhos das páginas do servidor web.**

Pág.	Tamanho	Pág.	Tamanho	Pág.	Tamanho	Pág.	Tamanho	Pág.	Tamanho
1	68 KB	6	582 KB	11	1,6 MB	16	2,4 MB	21	4,1 MB
2	162 KB	7	682 KB	12	1,6 MB	17	2,6 MB	22	5,9 MB
3	238 KB	8	1,2 MB	13	1,9 MB	18	2,9 MB	23	6,9 MB
4	245 KB	9	1,5 MB	14	2,0 MB	19	3,3 MB	24	7,9 MB
5	381 KB	10	1,5 MB	15	2,4 MB	20	3,5 MB	25	9,4 MB

A Tabela 3 apresenta o tamanho de cada uma das 25 páginas da aplicação. Três conjuntos de requisições de página ( $U$ ,  $E$ ,  $N$ ) foram criados para gerar tráfego de rede (diferentes quanto a demanda total) na máquina onde o servidor foi configurado. Cada conjunto é composto de 2500 requisições, e cada uma dessas requisita uma das 25 páginas do servidor. A diferença entre cada conjunto de requisições está na distribuição de probabilidade do número de página requisitado:

- $U$ : distribuição uniforme entre 1 e 25;
- $E$ : distribuição exponencial  $\text{Exp}(25/3)$  (quanto menor o número de uma página, maior a sua popularidade);
- $N$ : distribuição normal  $N(25/2, 25/6)$  (páginas com números intermediários são mais populares que páginas com números pequenos/grandes).

A infraestrutura do experimento consistiu em duas máquinas virtuais (MVs) (servidora e cliente) provisionadas sobre o monitor de máquinas virtuais Xen versão 4.4. Cada MV foi provisionada com: 1 VCPU, 512 MB de RAM, 20 GB de espaço em disco, uma interface de rede com largura de banda (LB) de envio limitada a 80 Mbps. O servidor de páginas web foi implantado na MV servidora. Na MV cliente executou-se a ferramenta `httperf`<sup>3</sup> para requisitar as páginas à servidora segundo os conjuntos  $U$ ,  $E$  e  $N$ .

Inicialmente foi caracterizado o tráfego para cada um dos conjuntos de requisições, e posteriormente as requisições foram classificadas usando as classes de fluxos obtidas. O objetivo foi o de avaliar a sensibilidade das estimativas a discrepâncias entre o tráfego usado para caracterização e o observado durante a classificação. A Tabela 4 apresenta a LB necessária estimada para cada par de conjunto de requisições; as linhas representam o conjunto usado para caracterização e as colunas o conjunto usado para classificação e estimativa. Os valores de cada coluna devem se aproximar do valor encontrado utilizando o mesmo conjunto de requisições para caracterizar o tráfego, calcular as frequências e estimar a LB necessária. Observa-se que, mesmo quando o tráfego usado na classificação foi diferente do usado na caracterização, as estimativas tiveram boa exatidão: o erro médio (considerando apenas os conjuntos diferentes) foi de 0,62%, e o erro máximo foi de 2,18%. Conclui-se que as classes de fluxos obtidas do processo de agrupamento de *flow records* representam bem as diferentes demandas das requisições feitas ao servidor de páginas, mesmo variando a popularidade de cada requisição considerada no agrupamento dos *flow records*.

**Tabela 4. Estimativas e erros para os pares de conjuntos de requisições.**

conjunto usado para caracterização	conjunto usado para classificação e estimativa		
	$U$	$E$	$N$
$U$	17,09 Mbps	10,30 Mbps (2,18%)	22,37 Mbps (0,54%)
$E$	17,13 Mbps (0,23%)	10,08 Mbps	22,24 Mbps (-0,04%)
$N$	17,11 Mbps (0,12%)	10,14 Mbps (0,60%)	22,25 Mbps

#### 4.2. Variação da Largura de Banda de Rede Reservada

Este experimento tem por objetivo avaliar a qualidade das estimativas dadas pelo modelo proposto quando a largura de banda (LB) reservada no período de monitoração varia entre valores subprovisionados, adequados e superprovisionados. Para este experimento foram criadas quatro cargas de trabalho, cada uma com páginas de tamanhos e taxas de requisição diferentes. As páginas das cargas foram implantadas em um servidor Apache com dois *virtual hosts*, um HTTP e outro HTTPS. A Tabela 5 apresenta o tamanho, a taxa de requisição e o protocolo de aplicação (HTTP ou HTTPS) de cada página de cada carga de trabalho. Realizou-se também experimentos com cargas de trabalho de outras aplicações – como FTP, SSH, e banco de dados SQL – mas seus resultados não são apresentados por serem semelhantes aos observados no presente experimento. Os tamanhos

<sup>3</sup><https://github.com/httperf/httperf>

das páginas foram escolhidos aproximando-se de tamanhos de páginas dos websites mais populares da Internet<sup>4</sup>. As frequências de requisições de cada página foram escolhidas para explorar diferentes distribuições de popularidade (e.g., páginas possuem popularidade uniforme, páginas menores são mais populares).

**Tabela 5. Cargas de trabalho para o segundo experimento.**

<i>Carga 1</i>				<i>Carga 2</i>			
<b>Pág.</b>	<b>Tamanho</b>	$\lambda$ (req/s)	<b>Protocolo</b>	<b>Pág.</b>	<b>Tamanho</b>	$\lambda$ (req/s)	<b>Protocolo</b>
1	750 KB	3	HTTP	1	100 KB	25	HTTP
2	1 MB	5	HTTP	2	1 MB	7	HTTP
3	1,75 MB	5	HTTP	3	4 MB	3	HTTP
4	400 KB	9	HTTPS	4	500 KB	10	HTTPS
5	800 KB	3	HTTPS	5	1,5 MB	5	HTTPS
6	1,6 MB	9	HTTPS	6	2 MB	5	HTTPS

<i>Carga 3</i>				<i>Carga 4</i>			
<b>Pág.</b>	<b>Tamanho</b>	$\lambda$ (req/s)	<b>Protocolo</b>	<b>Pág.</b>	<b>Tamanho</b>	$\lambda$ (req/s)	<b>Protocolo</b>
1	100 KB	20	HTTP	1	100 KB	20	HTTP
2	200 KB	25	HTTP	2	200 KB	20	HTTP
3	500 KB	20	HTTP	3	500 KB	15	HTTP
4	400 KB	25	HTTPS	4	400 KB	20	HTTPS
5	500 KB	15	HTTPS	5	500 KB	15	HTTPS
6	600 KB	15	HTTPS	6	600 KB	15	HTTPS

A infraestrutura do experimento consistiu em duas máquinas físicas (hospedeira e cliente) e uma MV (servidora). Cada máquina física possui 2 processadores Intel Xeon E5520 (4 cores por processador), 24 GB de RAM, 64 GB de armazenamento SSD e uma interface de rede Gigabit Ethernet. Ambas estão configuradas com o sistema operacional Debian 7, e adicionalmente, a máquina hospedeira executa o monitor de máquinas virtuais Xen versão 4.1. A MV servidora é virtualizada na hospedeira com 4 VCPUS, 4 GB de RAM, 4 GB de armazenamento SSD e LB da interface de rede variando de acordo com o cenário de teste do experimento. A máquina cliente faz as requisições das cargas de trabalho (pela ferramenta `httperf`) ao servidor Apache implantado na MV servidora.

Para cada carga de trabalho o modelo proposto foi executado completamente para sete cenários de reserva de LB, entre 40% e 160% da LB necessária (para determinar a LB necessária, a carga foi executada uma vez sem restrição da LB, ou seja, com 1 Gbps disponível). O tempo médio de execução do modelo foi de 7 segundos. Essa variação da LB reservada busca reproduzir os estados de subprovisionamento, provisionamento adequado e superprovisionamento e avaliar a qualidade das estimativas fornecidas pelo modelo proposto. As Figuras 4(a) e 4(b) apresentam as estimativas do modelo e o tempo médio de resposta para cada carga de trabalho em cada cenário de reserva de LB. Cada curva do gráfico representa os valores para uma carga de trabalho. Cada observação indica a LB estimada pelo modelo proposto ou o tempo de resposta (eixo y) quando executado sob determinada LB reservada (eixo x). O tempo de resposta reflete a duração de cada transação web, do início da requisição até o fim da resposta. Nos gráficos, os valores de LB são expressos como um percentual da LB necessária (quando a LB estimada ou reservada for igual à LB necessária esse percentual é de 100%).

<sup>4</sup><http://www.alex.com/topsites>

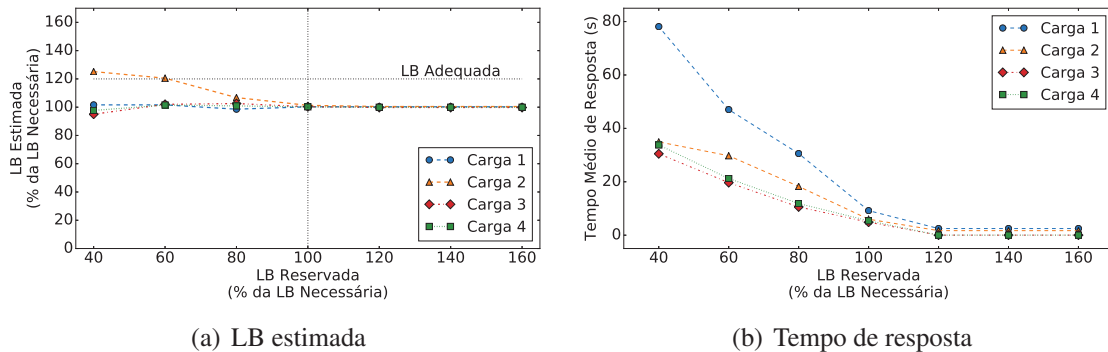


Figura 4. LB estimada e tempo de resposta em função da LB reservada.

Observa-se que as estimativas para os cenários onde a LB reservada é igual ou superior à LB necessária apresentam erros percentuais de pequena magnitude. Considerando o tempo médio de resposta, conclui-se que a LB mais adequada dentre as experimentadas está em 120% da LB necessária de cada carga, pois a partir deste ponto não há redução no tempo de resposta, mesmo com o aumento da capacidade. Constata-se também que os cenários de subprovisionamento (40%, 60% e 80%) resultam em erros maiores, mas ainda aceitáveis. Esse aumento era esperado visto que a rede fica saturada, o que por sua vez distorce o tráfego imprevisivelmente, favorecendo alguns tipos de fluxos em detrimento de outros. O maior erro observado, uma superestimação de aproximadamente 25%, ocorreu para a Carga 2, no cenário onde a LB reservada é igual a 40% da necessária. Mesmo no cenário mais desfavorável, o modelo consegue produzir uma estimativa útil para o provisionamento adequado da rede. Portanto, pode-se concluir que o modelo proposto é capaz de fornecer boas estimativas da LB necessária independentemente do nível atual de provisionamento da rede.

## 5. Conclusão

Neste trabalho foi proposto um novo modelo para estimação de largura de banda adequada para o provisionamento de máquinas virtuais em nuvens IaaS, considerando a perspectiva do cliente. O modelo é agnóstico a aplicação e baseia-se na monitoração de fluxos de pacotes nas interfaces de rede de uma máquina virtual. O processamento dos fluxos usando técnicas de aprendizado de máquina permite obter boas estimativas mesmo quando a rede está subprovisionada, um avanço em relação ao estado da arte. A avaliação experimental evidenciou que o modelo proposto é capaz de identificar acuradamente a demanda de rede com diferentes cargas de trabalho e de prover estimativas de largura de banda que são úteis para a tarefa de provisionamento de rede, independente da capacidade reservada durante sua execução. Em continuidade a este trabalho, está-se investigando a obtenção de previsões a partir de sequências de estimativas, e a incorporação do modelo proposto a um mecanismo de gerenciamento automático de escalabilidade.

## Agradecimentos

Os autores gostariam de agradecer à CAPES pelo auxílio financeiro e ao Laboratório de Processamento Paralelo e Distribuído (LabP2D) pela disponibilização dos recursos computacionais para a realização do trabalho. Alguns experimentos apresentados neste trabalho foram realizados usando o *testbed* Grid'5000, mantido por um grupo de interesse científico hospedado por Inria e inclui CNRS, RENATER, várias universidades e outras organizações (vide <https://www.grid5000.fr>).



## Referências

- [Anastasi et al. 2016] Anastasi, G. F., Coppola, M., Dazzi, P., and Distefano, M. (2016). QoS guarantees for network bandwidth in private clouds. *Procedia Computer Science*, 97:4–13.
- [Anhalt et al. 2010] Anhalt, F., Koslovski, G., and Primet, P. V.-B. (2010). Specifying and provisioning virtual infrastructures with HIPerNET. *Int. J. Network Management*, 20(3):129–148.
- [Ballani et al. 2011] Ballani, H., Costa, P., Karagiannis, T., and Rowstron, A. (2011). The price is right: Towards location-independent costs in datacenters. In *10th ACM Workshop on Hot Topics in Networks (HotNets)*.
- [Engelbrecht and van Greunen 2015] Engelbrecht, H. and van Greunen, M. (2015). Forecasting methods for cloud hosted resources, a comparison. In *Network and Service Management (CNSM), 2015 11th International Conference on*, pages 29–35. IEEE.
- [Gong et al. 2010] Gong, Z., Gu, X., and Wilkes, J. (2010). PRESS: Predictive elastic resource scaling for cloud systems. In *2010 International Conference on Network and Service Management*, pages 9–16. IEEE.
- [LaCurts et al. 2014] LaCurts, K., Mogul, J. C., Balakrishnan, H., and Turner, Y. (2014). Cicada: Introducing predictive guarantees for cloud networks. In *6th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 14)*.
- [Marcon et al. 2016] Marcon, D. S., Neves, M. C., Oliveira, R. R., Gaspar, L. P., and Barcellos, M. P. (2016). PredCloud: Providing predictable network performance in large-scale OpenFlow-enabled cloud platforms through trust-based allocation of resources. *Computer Communications*, 91:44–61.
- [Mell and Grance 2011] Mell, P. M. and Grance, T. (2011). The NIST definition of cloud computing. NIST SP 800-145, NIST, Gaithersburg, MD, United States.
- [Mogul and Popa 2012] Mogul, J. C. and Popa, L. (2012). What we talk about when we talk about cloud network performance. *ACM Computer Communication Review*, 42(5):44–48.
- [Moreira Jr and Obelheiro 2016] Moreira Jr, D. A. and Obelheiro, R. R. (2016). Previsão de demanda de recursos em nuvens IaaS usando séries temporais. In *VI Simpósio Brasileiro de Engenharia de Sistemas Computacionais (SBESC)*, João Pessoa, PB.
- [Nguyen et al. 2013] Nguyen, H., Shen, Z., Gu, X., Subbiah, S., and Wilkes, J. (2013). Agile: Elastic distributed resource scaling for infrastructure-as-a-service. In *Proceedings of the 10th International Conference on Autonomic Computing (ICAC 13)*, pages 69–82.
- [Pedregosa et al. 2011] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.
- [Pfitscher et al. 2013] Pfitscher, R. J., Pillon, M. A., and Obelheiro, R. R. (2013). Diagnóstico do provisionamento de recursos para máquinas virtuais em nuvens IaaS. *31o. Simpósio Brasileiro de Redes de Computadores (SBRC)*, pages 599–612.
- [Popa et al. 2012] Popa, L., Kumar, G., Chowdhury, M., Krishnamurthy, A., Ratnasamy, S., and Stoica, I. (2012). FairCloud: Sharing the network in cloud computing. In *Proc. ACM SIGCOMM*, pages 187–198.
- [Suleiman et al. 2012] Suleiman, B., Sakr, S., Jeffery, R., and Liu, A. (2012). On understanding the economics and elasticity challenges of deploying business applications on public cloud infrastructure. *Journal of Internet Services and Applications*, 3:173–193.